ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Laboratorio di

Sicurezza Informatica

Crittografia moderna

Marco Prandini

Dipartimento di Informatica – Scienza e Ingegneria

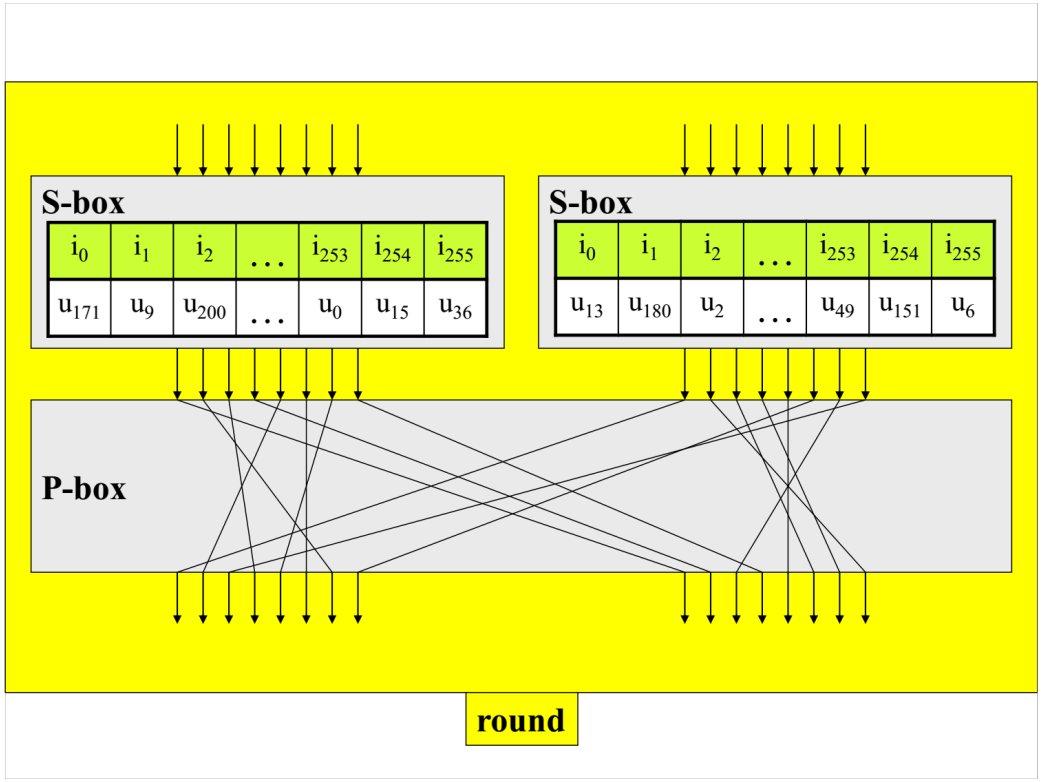
Cifrari a blocchi

Partendo dai cifrari classici:

- Osservazioni di base sull'input
  - È utile ridurre a priori la riconoscibilità statistica dei simboli dell'alfabeto
    - Aumentandone il numero (frequenza media = 1/N) → facile se prendo come "lettera" un blocco di 8 bit, ma anche 16, 32, 64...
    - Rendendoli equiprobabili (compressione)
- Osservazioni di base sull'algoritmo
  - Ogni operazione di sostituzione e trasposizione aumenta la confusione e la diffusione

I cifrari composti

- Un *round* sostituisce e traspone
- Tanti round incrementano l'effetto



Cifrari a blocchi

- La parte difficile è implementare E e D “modularmente” per poter lavorare liberamente sul numero di round
- Cifrari di Feistel
- Standard storico: DES (National Bureau of Standards degli U.S.A in collaborazione con IBM, pubblicato nel 1977, blocchi di 64 bit, chiave di 56 bit)

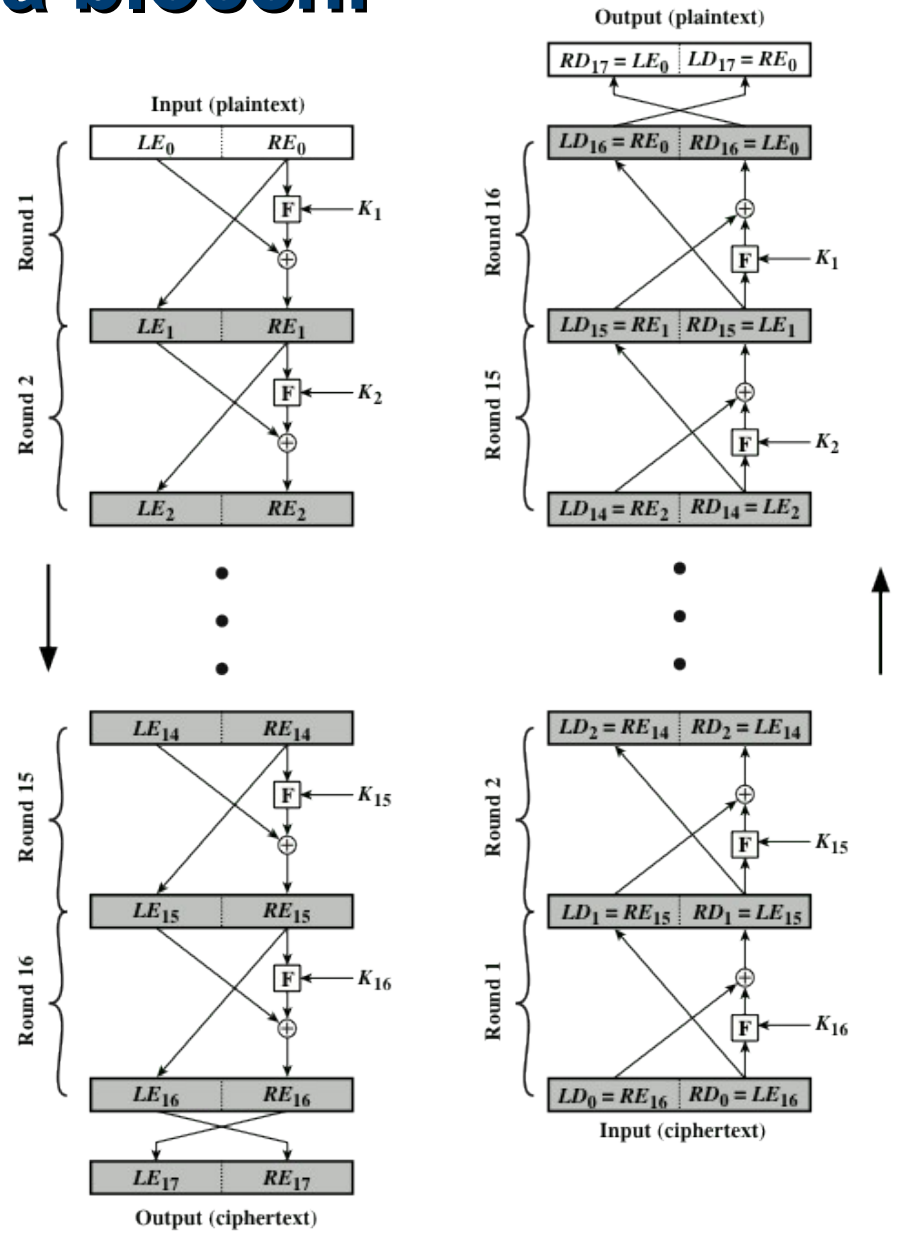


Figure 3.3 Feistel Encryption and Decryption (16 rounds)

AES

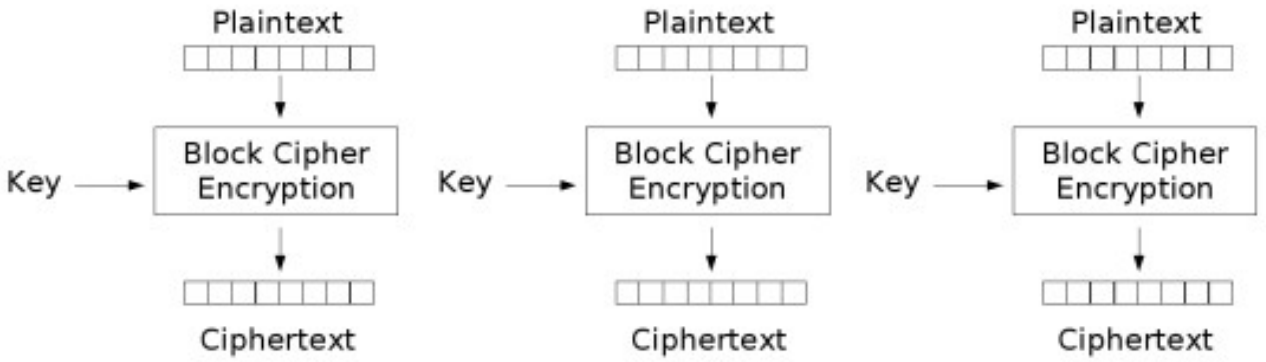
- Standard attuale: FIPS 197 “Advanced Encryption Standard” (Rijndael)
  https://csrc.nist.gov/publications/detail/fips/197/final
  - Interessante il processo di selezione

    https://csrc.nist.gov/projects/cryptographic-standards-and-guidelines/archived-crypto-projects/aes-development

- Non usa la struttura di Feistel ma l’aritmetica dei campi finiti
- Blocchi di 128 bit
- Può utilizzare chiavi di lunghezza diversa
  - 128 bit
  - 192 bit
  - 256 bit

I modi di operazione

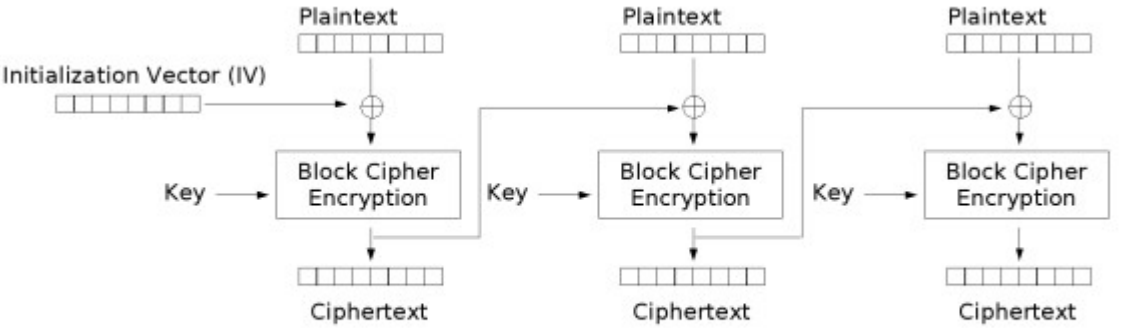
- Cifrare blocco per blocco è male
  - Stesso plaintext = stesso ciphertext → analisi facilitata
  - Modifica a un blocco = altri blocchi inalterati → integrità non protetta



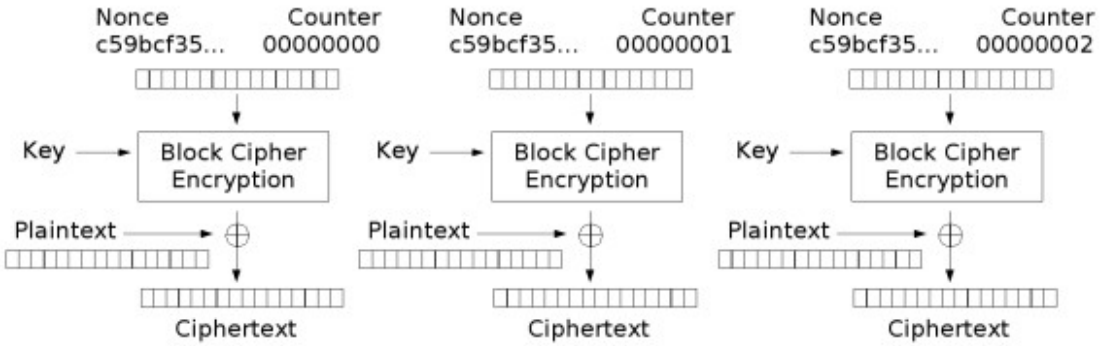
Electronic Codebook (ECB) mode encryption

I modi di operazione

- Soluzione:
  - Cifrare un blocco modificandolo col contributo del blocco cifrato precedente
    - Cipher Block Chaining (CBC)
    - Cipher Feedback (CFB)
    - Output Feedback (OFB)
  - Realizzare l'equivalente a blocchi di un cifrario a flusso
    - Counter (CTR)



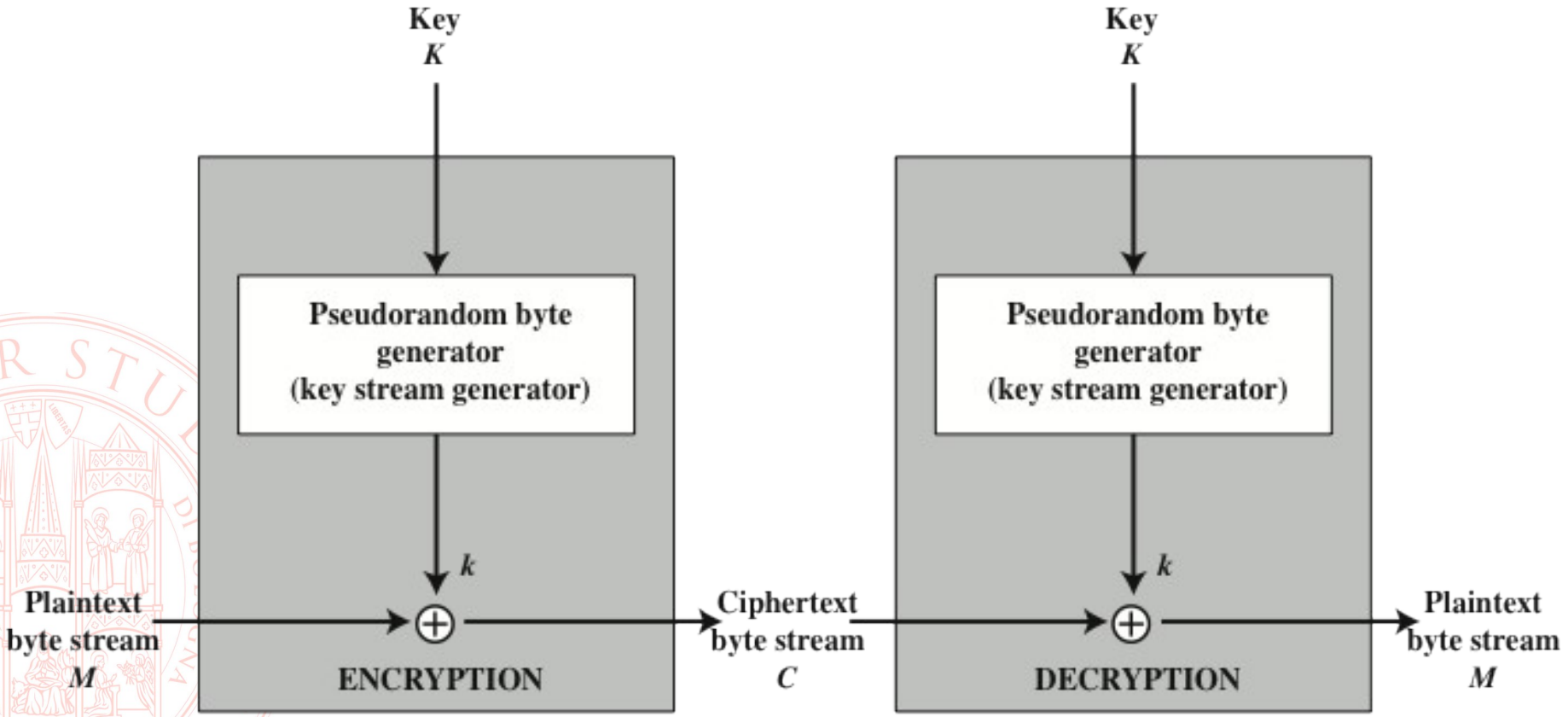
Cipher Block Chaining (CBC) mode encryption



Counter (CTR) mode encryption

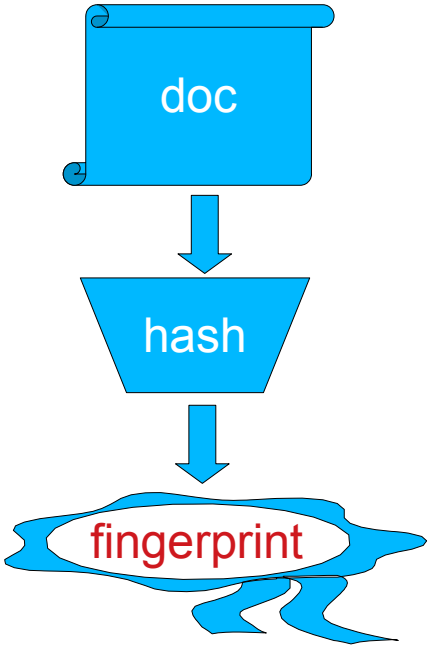
Cifrari a flusso

- One-Time Pad con alfabeto insignificante: solo 0 e 1 → Analisi statistica delle frequenze inapplicabile
- *Flusso di chiave* = generazione sequenza casuale
  - Seme = chiave condivisa



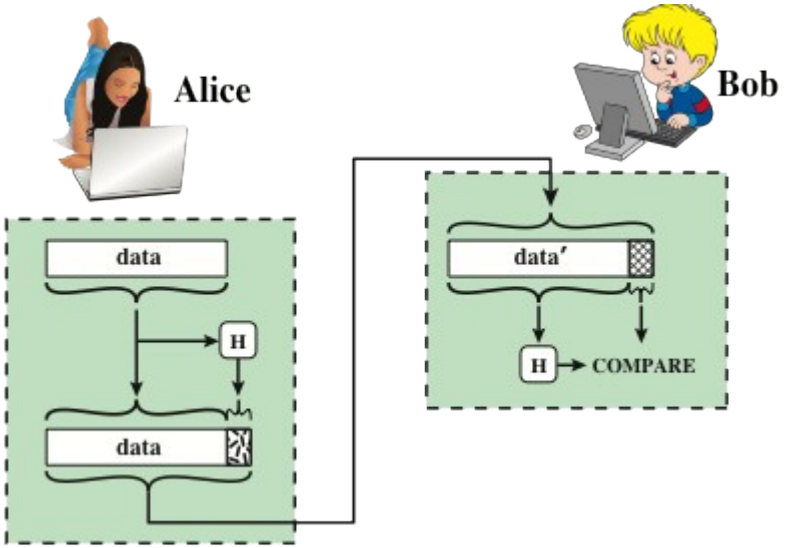
Funzioni hash

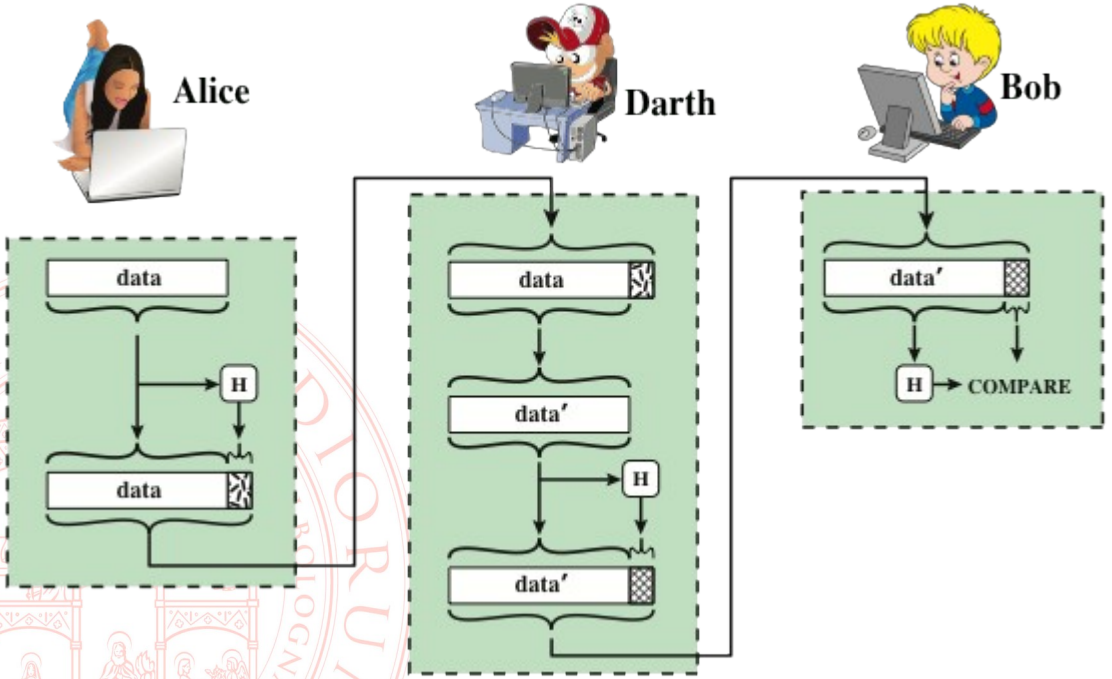
- Gli stessi principi dei cifrari a blocchi possono essere usati senza chiave per ottenere “impronte digitali” compatte di documenti di dimensione arbitraria
- Fingerprint:
  - dimensione fissa (f. non biunivoca)
  - f. pubblica, senza chiave
- Funzioni hash crittografiche – robuste se:
  1) Non si può trovare un documento che abbia un fingerprint prefissato (proprietà di unidirezionalità, o **one-way**)
  2) Non si può trovare una coppia di documenti con lo stesso fingerprint (proprietà di assenza di collisioni, o **collision-free**)



Utilità delle funzioni hash

- Integrità (checksum)
  - ✔ Ok contro alterazioni accidentali

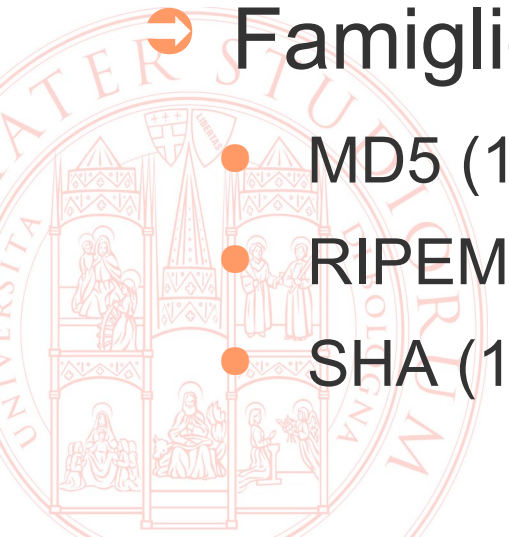




- ⊖ Man-in-the-middle attack!
- Protezione necessaria
  - Canale sicuro
  - Altro?
- Autenticazione?
  - Manca elemento univoco dell'autore

Attacchi a one-way property

- Ricerca di difetti nell'algoritmo
  - Improbabile
- Forza bruta
  - Generare documenti a caso e vedere se hanno la fingerprint cercata
  - Tempo crescente esponenzialmente con la lunghezza dell'impronta
- Famiglie più diffuse
  - MD5 (128 bit), MD6 (fino a 512 bit)
  - RIPEMD (128, 160,320 bit)
  - SHA (160, 224,256,384,512) / SHA-3 (arbitraria)



Attacchi a collision-free

- Ricerca di difetti nell'algoritmo
  - Trovati! SHA-1 (2005), MD5 (2008)
- Birthday attack (paradosso del compleanno)
  - Dimensione di un gruppo per avere probabilità >50% che ci sia un compleanno in una data specifica?
    - $P_{non\_compleanno} = \frac{364}{365}$; in gruppo di N, $P_{nessun\_compleanno} = (\frac{364}{365})^N$
    - P<0.5 → N>253
  - Dimensione di un gruppo per avere probabilità >50% che due membri compiano gli anni lo stesso giorno?
    - Per singola coppia di persone, stesso calcolo
    - Ma vale per N coppie! Con M persone compongo $\frac{M*(M-1)}{N}$ coppie
    - Bastano circa $M=\sqrt{2N}$ persone = 23
  - Per un hash di *m* bit, la dimensione del set è 2^m
  - Per trovare una coppia di documenti con lo stesso hash bastano **$2^{m/2}$** tentativi

Una lettera in 2^{37} varianti

Dear Anthony,

{This letter is / I am writing} to introduce {you to / to you} {Mr. / --} Alfred {P. / --}

Barton, the {new / newly appointed} {chief / senior} jewellery buyer for {our / the}

Northern {European / Europe} {area / division}. He {will take / has taken} over {the / --}

responsibility for {all / the whole of} our interests in {watches and jewellery / jewellery and watches}

in the {area / region}. Please {afford / give} him {every / all the} help he {may need / needs}

to {seek out / find} the most {modern / up to date} lines for the {top / high} end of the

market. He is {empowered / authorized} to receive on our behalf {samples / specimens} of the

{latest / newest} {watch and jewellery / jewellery and watch} products, {up / subject} to a {limit / maximum}

of ten thousand dollars. He will {carry / hold} a signed copy of this {letter / document}

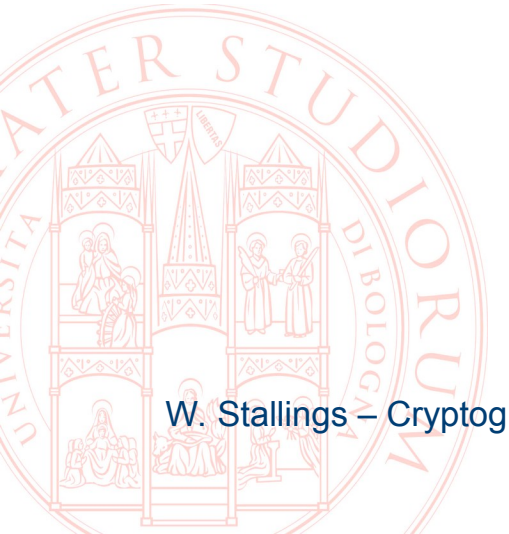
as proof of identity. An order with his signature, which is {appended / attached}

{authorizes / allows} you to charge the cost to this company at the {above / head office}

address. We {fully / --} expect that our {level / volume} of orders will increase in

the {following / next} year and {trust / hope} that the new appointment will {be / prove}

{advantageous / an advantage} to both our companies.



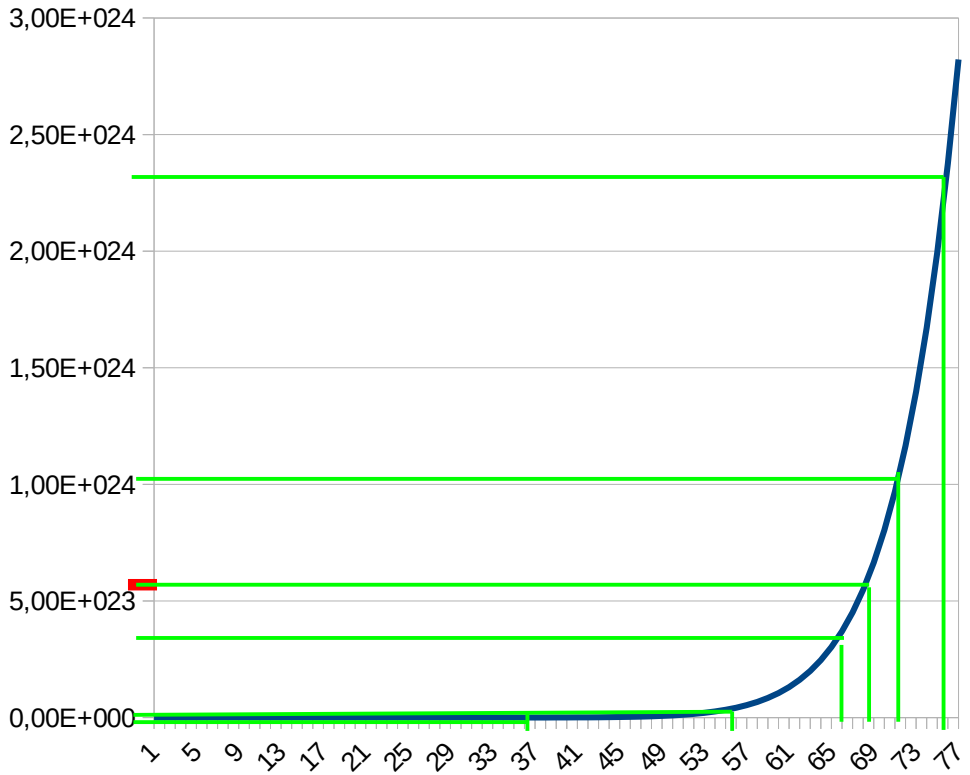
Altri attacchi

- Lenght extension
  - Noto $H(m_1)$ e la lunghezza di m_1
  - Senza conoscere m_1
  - Scelto un m_2 dall'attaccante
  - È possibile calcolare $H(m_1 \,||\, m_2)$
- Principali vulnerabili:
  - MD5, SHA-1, RIPEMD-160, SHA-256, SHA-512
- Principali resistenti:
  - SHA-3, varianti troncate di SHA-2

Problemi difficili e trabocchetti

- Funzioni *pseudo-unidirezionali*
  - Operazioni facili in un verso e (speriamo) computazionalmente infattibili nell'altro
  - A meno di conoscere un segreto
- Fattorizzazione di grandi numeri
- Molte operazioni in aritmetica modulare
  - Numeri interi
  - Come risultato di un'operazione si prende il resto della divisione per un *modulo* fisso

Intuitivamente



y=x^13

Su R, se non conosco l'inversa di una funzione “regolare”, mi avvicino per approssimazioni successive (es. bisezione)

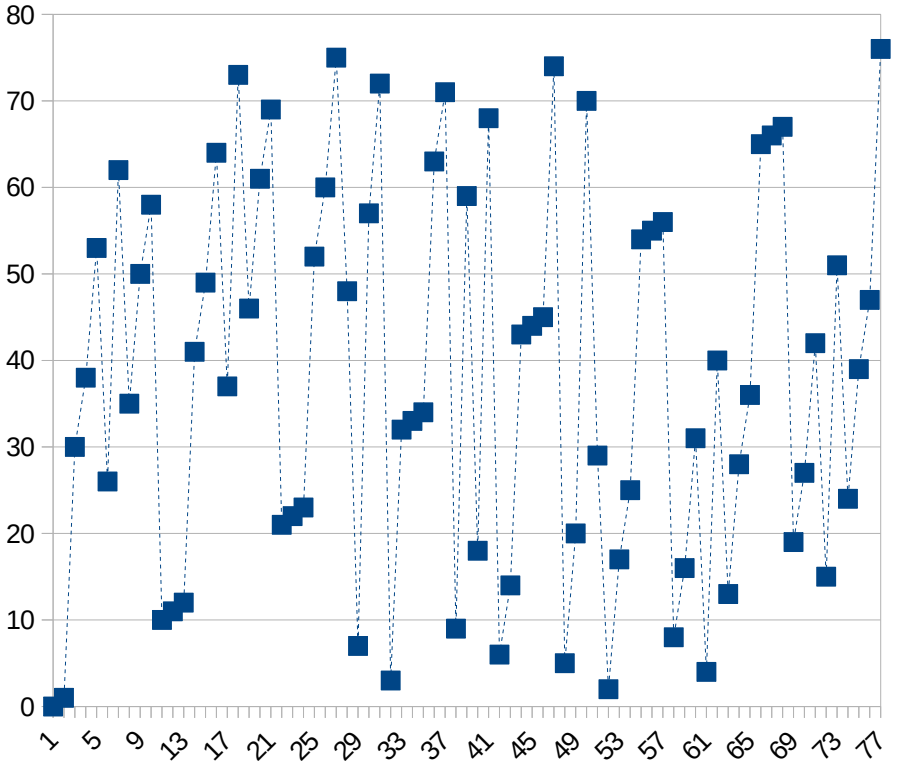
Per una funzione monotona, si parte dagli estremi del dominio, e si valuta la funzione nel punto medio del dominio.
In questo esempio, valutiamo la funzione per {0,76} → {38,76} → {57,76} → {66,5,76} → {66,5,71,25}

Per x=68.875 otteniamo il risultato

Intuitivamente

y=x^13 mod 77

Su Z_{77}, (il campo di Galois con 77 numeri, in cui le operazioni si effettuano modulo 77) l'effetto di riduzione modulare rende estremamente irregolare la funzione → non è possibile una ricerca efficiente

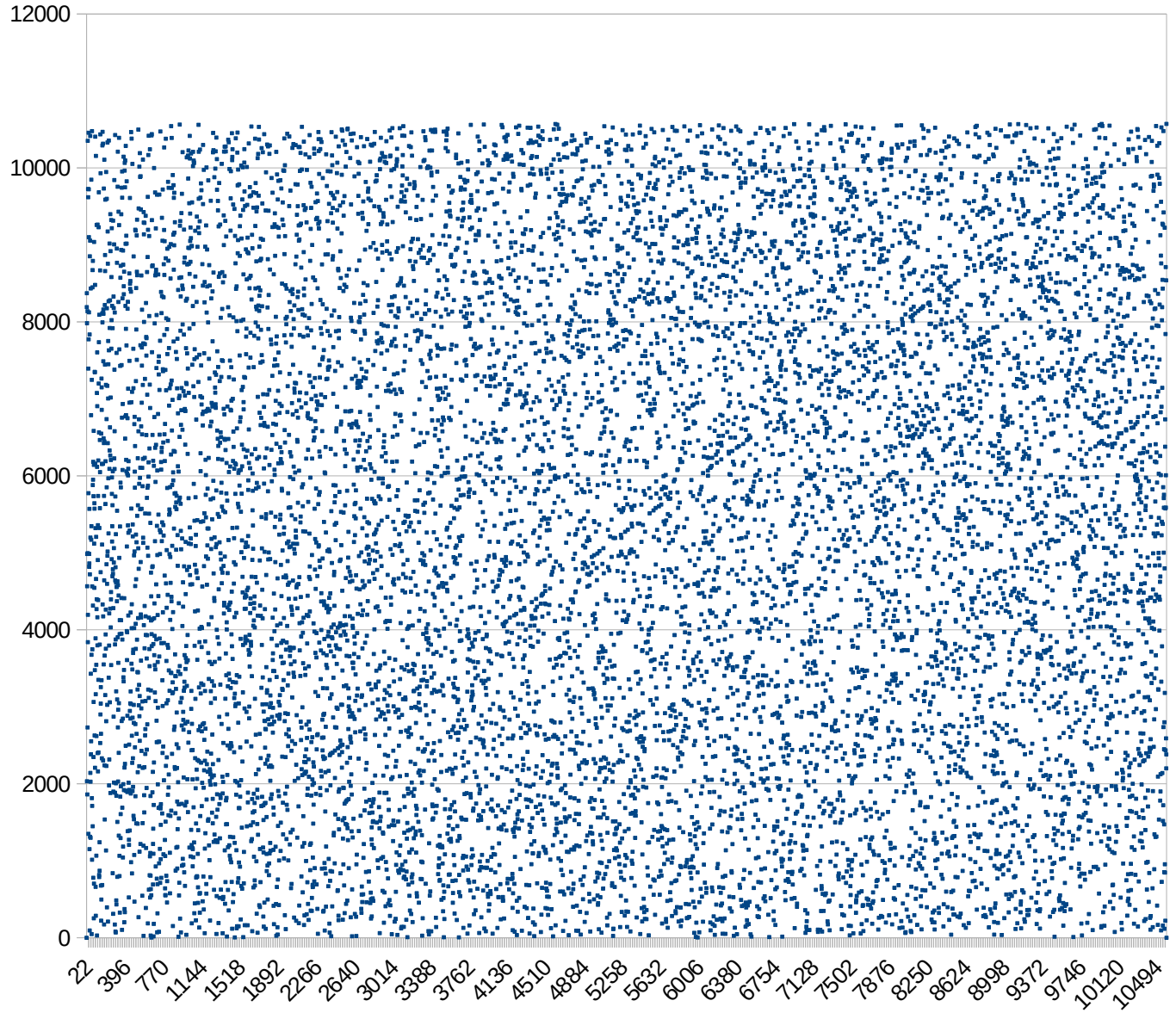


Crittografia asimmetrica: RSA (1977)

- Generazione delle chiavi:
  1. si scelgono due numeri primi p e q
  2. il modulo viene calcolato come $n = p \cdot q$
  3. si sceglie a caso un numero d e si calcola un numero e tale che $e \cdot d \bmod (p-1)(q-1) = 1$
  - Facile solo conoscendo p e q, che vengono poi dimenticati

- La chiave pubblica è (e, n), la chiave privata (d, n)
- Cifratura: $c = m^e \bmod n$
- Decifrazione $m = c^d \bmod n$

$c=m^e \bmod n$ visivamente



e=13
P=97
Q=109

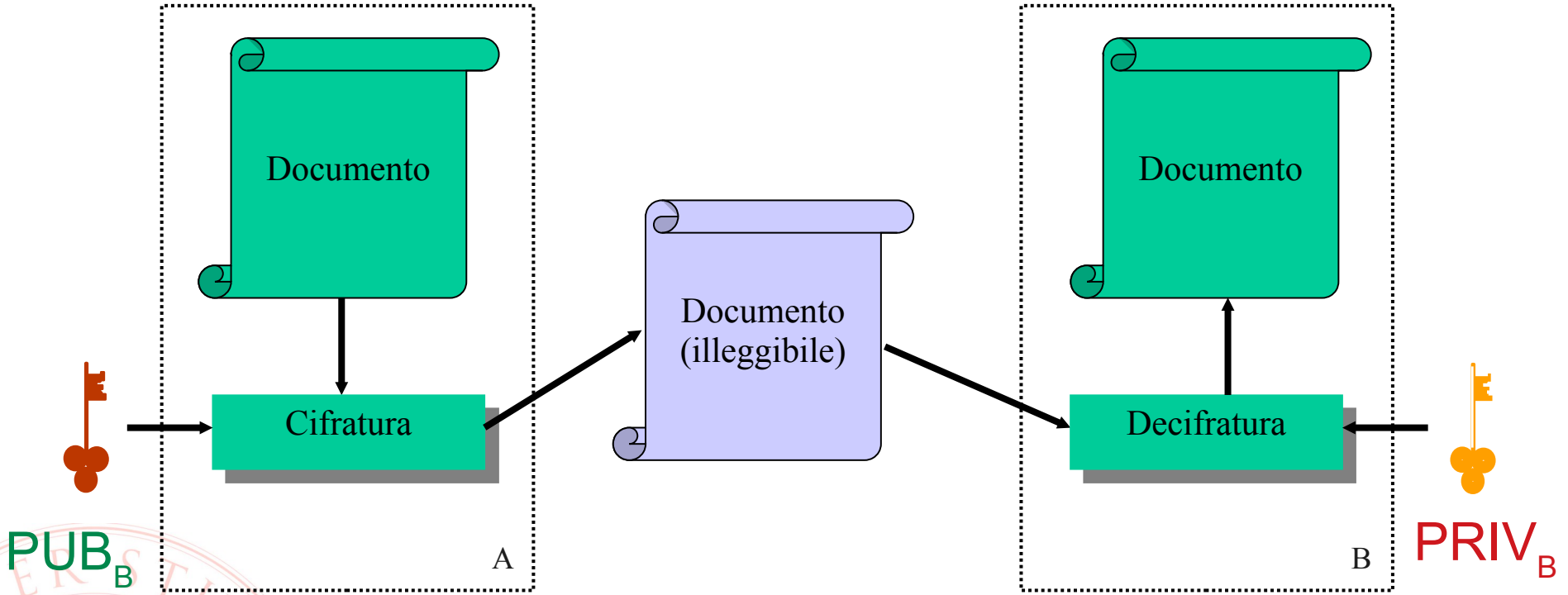
Robustezza

- Non ci sono modi efficienti noti di invertire l'esponenziale modulare
  - Complessità assimilabile a forza bruta
- Ci sono algoritmi "quasi efficienti" per fattorizzare il modulo
  - General Number Field Sieve, sub-esponenziale
  - Contromisura: **moduli grandi (oltre 2048 bit)**
- Trappole
  - Non è dimostrabile che non esistano algoritmi classici efficienti (ma nessuno ha idea di come trovarli)
  - Quantum computing
  - Implementazioni troppo efficienti
    - Spesso si sceglie *e* con pochi "1" (es. 3, 17, 65537)
    - Se troppo piccolo, m^e non "trabocca" da n!

Vantaggi della c. asimmetrica

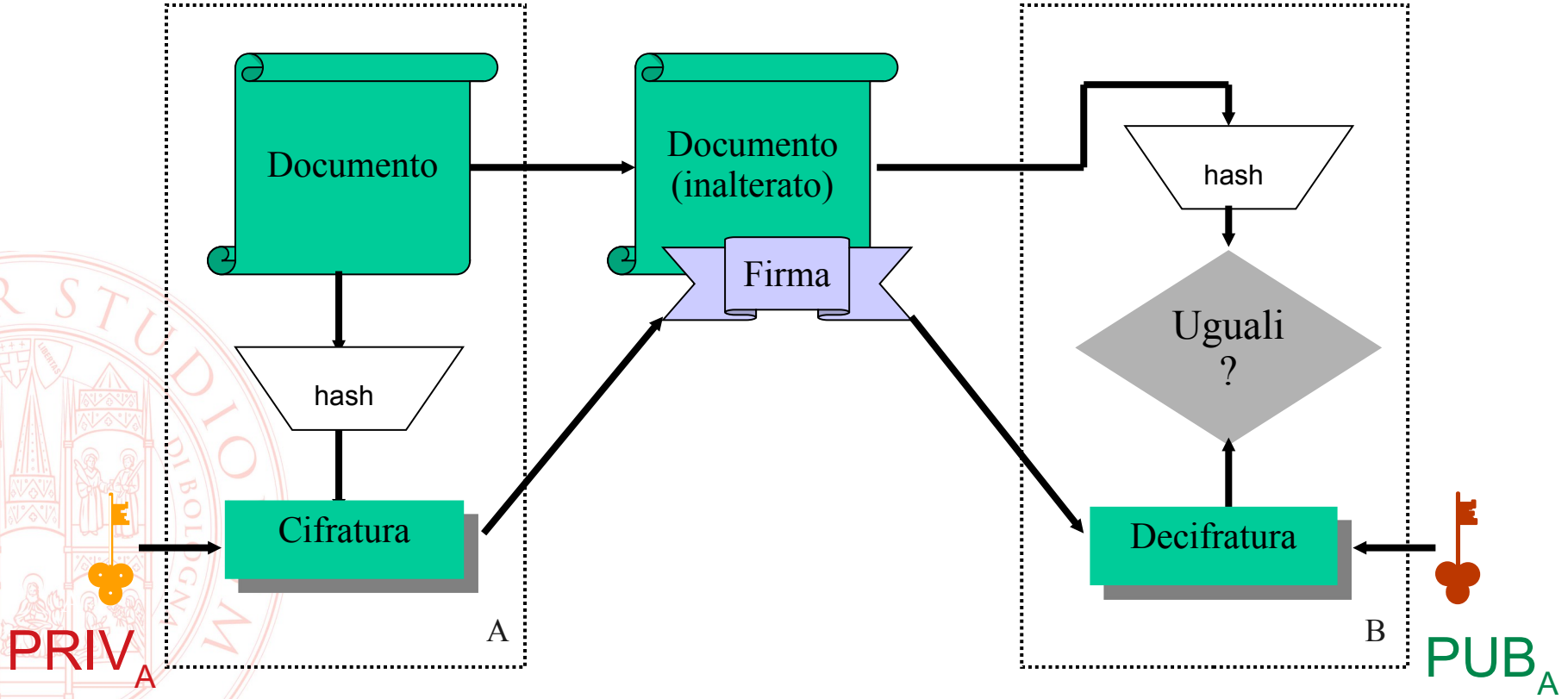
- Per la riservatezza
  - È un cifrario a blocchi, di sostituzione, con dimensioni enormi
    - No forza bruta
    - No analisi statistica dell'"alfabeto" (salvo casi particolari)
  - Le chiavi usate per cifrare e decifrare sono diverse e dalla chiave pubblica non è derivabile la chiave privata
    - La chiave pubblica può essere distribuita
    - Chiunque può usarla per cifrare
    - La chiave privata corrispondente è l'unica che può decifrare
- La chiave privata è specifica di un solo utente quindi utile anche per *autenticare*

C. asimmetrica per la riservatezza



C. asimmetrica per l'integrità e l'autenticità

- Soluzione del problema dell'uomo nel mezzo visto per gli hash: cifrare il fingerprint con la chiave privata
  - Verifica corretta solo se tutto inalterato → integrità
  - Verifica corretta con PUB_A solo se la firma era stata prodotta con $PRIV_A$ → autenticità (se posso fidarmi che PUB_A sia davvero di A!)

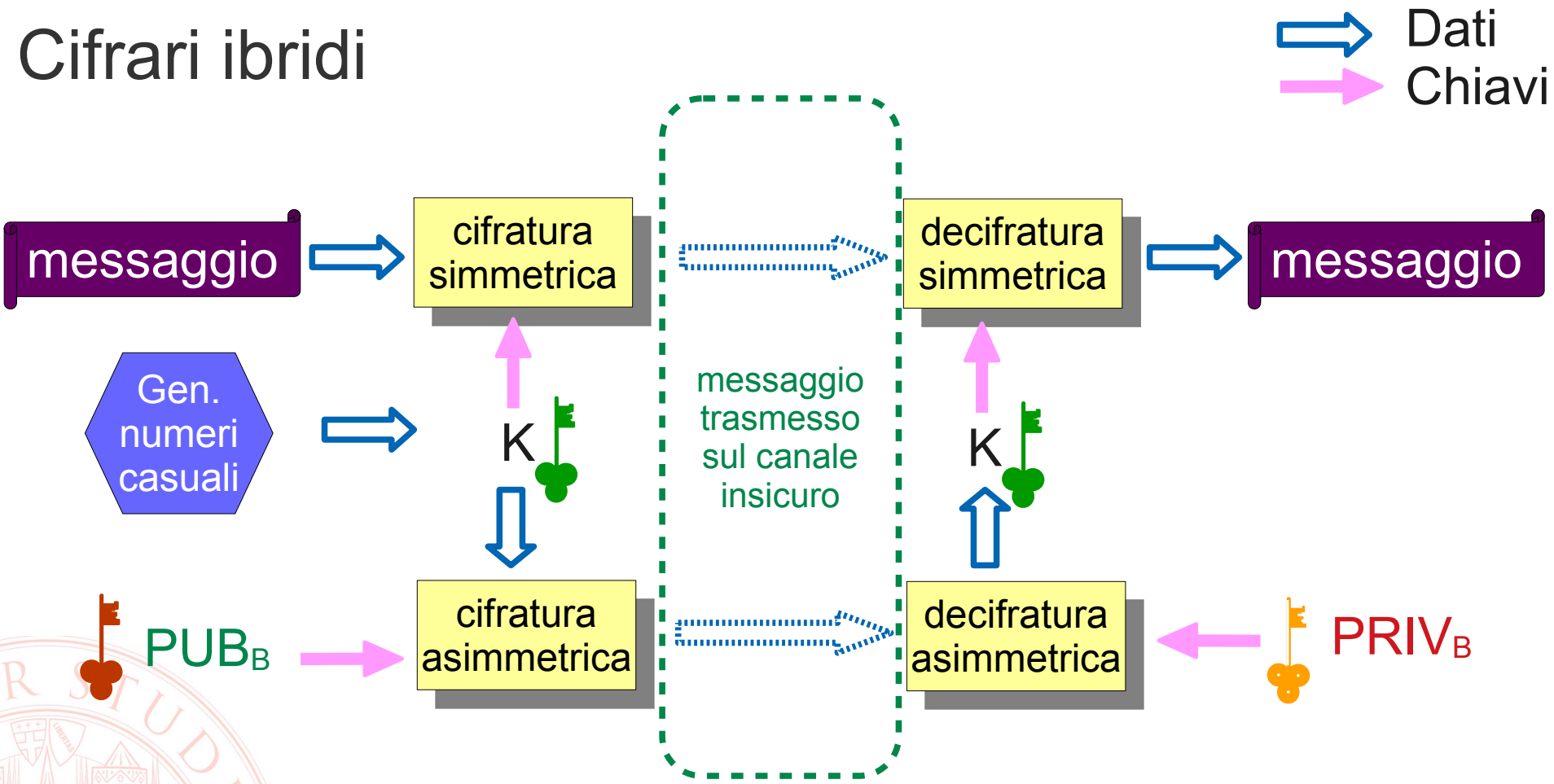


C. asimmetrica - pregi e difetti

- Grandi vantaggi:
  - distribuzione delle chiavi
  - utilità per tutte le proprietà di sicurezza
- Punti deboli:
  - Prestazioni (5-10 volte più lento di AES)
    - Sistemi ibridi
  - alcuni attacchi specifici (known plaintext)

Aggiungere prestazioni e flessibilità

Cifrari ibridi



Più destinatari = un solo messaggio cifrato &
più copie di K cifrate con la chiave pubblica di ognuno

Un flash su Quantum Computing

- Crittografia simmetrica e hash: nessun vero problema
  - Algoritmo di Grover: complessità ≈ sqrt(dimensione spazio di ricerca)
  - Compensato raddoppiando la lunghezza delle chiavi o delle fingerprint
- Crittografia asimmetrica basata su logaritmi discreti: spacciata
  - Algoritmo di Shor fattorizza in tempo polinomiale
  - Servono molti più qubit e gate di quanto ora realizzabile
    - Rischio sul lungo periodo
    - Ma quando accadrà, crollo istantaneo!
- Post-quantum cryptography
  - Algoritmi già talmente avanzati da essere standardizzati
  - https://en.wikipedia.org/wiki/Post-quantum_cryptography